Lustre File System for the Cray XD1™ System

S–2452–131

New Features

Lustre File System for the Cray XD1™ System S–2452–131

This manual provides improved explanatory material and examples.

Record of Revision

Version	*Description*
1.3.1	October 2005 Supports the Cray XD1 release 1.3.1.
1.3	July 2005 Supports the Cray XD1 release 1.3.
1.2.1	May 2005 Supports the Cray XD1 release 1.2.

Contents

Page

Figures

Tables

Preface

The information in this preface is common to Cray documentation provided with this software release.

Accessing Product Documentation

With each software release, Cray provides books and man pages, and in some cases, third-party documentation. These documents are provided in the following ways:

CrayDoc

The Cray documentation delivery system that allows you to quickly access and search Cray books, man pages, and in some cases, third-party documentation. Access this HTML and PDF documentation via CrayDoc at the following locations:

- The local network location defined by your system administrator
- The CrayDoc public website: `docs.cray.com`

Man pages

Access man pages by entering the `man` command followed by the name of the man page. For more information about man pages, see the `man`(1) man page by entering:

```
% man man
```

Third-party documentation

Access third-party documentation not provided through CrayDoc according to the information provided with the product.

Conventions

These conventions are used throughout Cray documentation:

Convention	Meaning
`command`	This fixed-space font denotes literal items, such as file names, pathnames, man page names, command names, and programming language elements.
variable	Italic typeface indicates an element that you will replace with a specific value. For instance, you may replace *filename* with the name `datafile` in your program. It also denotes a word or concept being defined.
`user input`	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. Output is shown in nonbold, fixed-space font.
[ ]	Brackets enclose optional portions of a syntax representation for a command, library routine, system call, and so on.
. . .	Ellipses indicate that a preceding element can be repeated.
`name`(N)	Denotes man pages that provide system and programming reference information. Each man page is referred to by its name followed by a section number in parentheses. Enter: `% man man` to see the meaning of each section number for your particular system.

Reader Comments

Contact us with any comments that will help us to improve the accuracy and usability of this document. Be sure to include the title and number of the document with your comments. We value your comments and will respond to them promptly. Contact us in any of the following ways:

E-mail:
`docs@cray.com`

Telephone (inside U.S., Canada):
1–800–950–2729 (Cray Customer Support Center)

Telephone (outside U.S., Canada):
+1–715–726–4993 (Cray Customer Support Center)

Mail:
Software Publications
Cray Inc.
1340 Mendota Heights Road
Mendota Heights, MN 55120–1128
USA

Cray XD1 Support

Obtain support for the Cray XD1 product in either of the following ways:

Telephone:
1–888–279–2729 (Cray XD1 Customer Support Center)

Through the CRInform website:
`http://crinform.cray.com/xd/`

Note: Use the contact information provided here if you have a support agreement with Cray. If, however, you have a support agreement with a third-party organization that is a Cray channel partner, contact that organization instead: do not contact Cray directly.

Lustre File System [1]

For Cray XD1 systems, the Lustre file system is an optional product. Lustre version 1.4 is a scalable, high-performance, POSIX-compliant file system that uses the Linux `ext3` file system for backend storage.

Additional general information about Lustre can be found at `http://www.lustre.org/documentation.html` and `http://www.clusterfs.com/faq.html`.

1.1 How Lustre Works

The Lustre file system consists of software subsystems, storage, and an associated network. You create a Lustre configuration file that defines the file system characteristics.

1.1.1 Lustre Software

There are three software components of Lustre that can run across all or selected nodes of the Cray XD1 system.

- Clients

  Clients are services or programs that access the file system. You do not set up clients as part of the Lustre setup, but you do have to specify the nodes on which they will run.

- Object storage target (OST)

  You can configure one or more *object storage target (OST)*s, which are software interfaces to backend storage volumes. The OSTs handle file data and enforce security for client access. Object storage targets provide a networked interface to other object storage. The client performs parallel I/O operations across multiple OSTs.

  You configure the characteristics of the OSTs as part of the Lustre setup.

- Metadata server

  The *metadata server (MDS)* owns and manages information about the files in the Lustre file system. It handles namespace operations such as file creation, but it does not contain any file data. It stores which file is located on which OST, how the blocks of files are striped across the OSTs, the date and time the

file was modified, and so on. The MDS is consulted whenever a file is opened or closed, and it may be referenced during I/O to get the block layout on the OSTs. Because file namespace operations are done by the MDS, they do not impact operations that manipulate file data.

The metadata server transforms client requests into journaled, batched metadata updates on persistent storage. The MDS can batch large numbers of requests from a single client, such as when a client grows a writeback cache, or it can batch large numbers of requests generated by different clients, such as when many clients are updating a single object.

You configure the characteristics of the MDS as part of the Lustre setup.

Each pair of subsystems acts according to protocol:

1. MDS-Client: The MDS interacts with the client for metadata handling such as the acquisition and updates of inodes, directory information, and security handling.
2. OST-Client: The object storage target interacts with the client for file data I/O, including the allocation of blocks, striping, and security enforcement.
3. MDS-OST: The MDS and OST interact to preallocate resources and perform recovery.

Lustre layout is shown in Figure 1.

Note: You can have one or more instances of the Lustre file system. A given MDS, OSTs, and associated clients constitute one Lustre file system. If you want to create another instance of Lustre, you must configure it on different nodes.

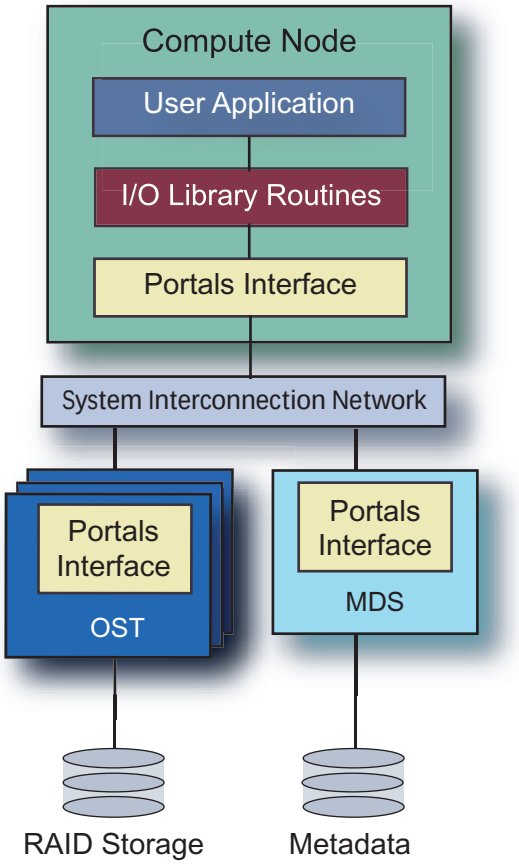

Figure 1. Layout of Lustre File System

The Lustre framework allows you to structure your file system installation to match your data transfer requirements. One MDS plus one or more OSTs make up a single instance of Lustre and are managed together. Client nodes mount the Lustre file system over the network and access files with POSIX file system semantics. Each client mounts Lustre, uses the MDS to access metadata, and performs file I/O directly through the OSTs.

1.1.2 Storage

Lustre accesses the logical units (LUNs) of storage attached to Cray XD1 system nodes `-5` and `-6`, which have two PCI slots. For more information about storage, see *Integrating Storage Devices with the Cray XD1 System*.

Warning: Do not configure your storage with LUNs greater than 2 TB. Additional storage in the LUN is ignored. Thus, the maximum size for a Lustre volume is 2 TB times the number of OSTs.

1.1.3 Network

You specify the Lustre network in the Lustre configuration file. Specify the RapidArray (ra) network. It is identified by the `--nettype` option of the `lmc` command (see Section 1.5, page 10).

1.1.4 Other Lustre Concepts

This section describes other Lustre concepts that are helpful to understand before you undertake the task of setting up a Lustre file system.

1.1.4.1 Lustre Configuration File

Each instance of a Lustre file system is generated by an XML configuration file that describes the characteristics of the file system: MDS, OST, clients, network, and storage specifications. A copy of the configuration file resides on the MDS and OST nodes for that instantiation of the Lustre file system. The first step in setting up the file system is to create this file.

A Lustre shell script (often called `config.sh`) provides input to generate the configuration file (located by convention in `/etc/lustre/config.xml`); you modify your shell script to include the Lustre `/usr/sbin/lmc` commands that specify the characteristics of the file system. For example, you can define the networking at each node, the location, name, and size of the components of Lustre, and the mount point of the file system. When you run the shell script, the configuration file is built.

After you have created a configuration file, you run the `lconf` command to start Lustre services on the components allocated in the file. You do not have to create a configuration file each time you start your system. If it has not changed, you can use the existing file for subsequent start up.

Clients mount Lustre like an NFS-exported file system. You mount Lustre to invoke the client.

Section 1.5, page 10 contains a description of how to build the configuration file to set up Lustre. Appendix B, page 25 provides a sample file. For command information, see the Appendix A.

1.1.4.2 Logical Object Volume

A logical object volume (LOV) groups a set of OSTs with an MDS and allows you to reference them as a unit. You configure the LOV with the `--lov` option of the `lmc` commands that you include in the shell script (see Section 1.5, page 10).

Warning: Do not associate an MDS with more than one LOV. This will cause file system corruption.

1.1.4.3 Generic Zero-configuration Client

A generic zero-configuration client acts as a wild card to allow any client node, instead of only the clients you specify, to mount a Lustre file system. It gives you the flexibility to add nodes as Lustre clients without copying your Lustre configuration file to each one. You can configure Lustre with or without a zero-configuration client.

To create a zero-configuration client, you must identify it in your configuration file and reformat your file system with the new configuration (see Section 1.5, page 10).

1.1.4.4 Object Storage Server

An *object storage server (OSS)* is a node that hosts OSTs. Each OSS node has fibre channel connections to a DataDirect or Engenio RAID couplet. For a discussion of Lustre layout, see Section 1.4.2, page 8.

1.1.5 Common Lustre Commands

Administrator-run commands that configure Lustre are shown in Table 1.

Table 1. Common Lustre Commands

Command	Function
`lmc`	Lustre make configuration; describes site-specific components; creates Lustre configuration file; rerun to change configuration.
`lconf`	Lustre configuration tool; starts and stops Lustre services using data in Lustre configuration file.
`lfs`	Reads and sets striping patterns across OSTs.

For more information about the tools, see Appendix A, page 19.

1.2 Location of Lustre Kernel Modules

Lustre kernel module files are loaded as part of the system boot and are stored in:

`/lib/modules/`*`version`*`/kernel/fs/lustre/`

`/lib/modules/`*`version`*`/kernel/net/lustre/`

You can determine the Linux kernel *version* by issuing the `uname -r` command.

By convention, configuration files are stored in `/etc/lustre`.

For information about installing Lustre, see *Cray XD1 System Administration*, "System and Site Configuration."

1.3 Setting the Deadline I/O Scheduler

For good performance, the deadline I/O scheduler must be set for Lustre nodes. Active Manager can set this parameter for all nodes with no ill effect to non-Lustre nodes. To set the deadline I/O scheduler, add the following line to the `/opt/pce/lib/amserver.properties` file:

```
EXTRA_BOOT_PARAMS=elevator=deadline
```

Restart Active Manager.

If you are experiencing problems. contact your Cray Service representative.

1.4 Configuring the Lustre File System

You can create and mount more than one instance of the Lustre file system. Follow the configuration process for each instance.

Each Lustre configuration is stored in an XML configuration file (see Section 1.1.4.1) that contains the network, storage, OST, MDS, and client specifications.

1.4.1 Striping

Striping is the process of distributing data from a single file across more than one device. To improve file system performance, you can stripe files across several or all OSTs.

You can specify values to create a system-wide default striping pattern with the `lmc` command when you create the configuration file, or you can run the `lfs` command to create and stripe files at a later time (see Appendix A, page 19). You can choose:

- The number of OSTs that each file is striped across. You can stripe across any number of OSTs, from a single OST to all available OSTs.
- The OST that will contain the first stripe.
- The number of bytes in each stripe.
- To override striping for individual files.

Warning: Striping increases the rate that data files can be read or written. However, if storage is not configured with redundancy, such as is available in a RAID configuration, reliability decreases as the number of stripes increases. Damage to an object storage target can cause loss of some data in many files.

Striping is configured when the LOV or an individual file is created. Cray recommends that unless your application suggests otherwise:

- Stripe files across all OSTs.
- Choose a stripe size of 1 MB (1048576 bytes).

  You can increase the stripe size by powers of two, but there is rarely a need to configure a stripe size of greater than 2 TB.

  Note: Cray recommends that you do not choose a smaller stripe size, even for files with writes that are smaller than the stripe size. The system caches smaller writes. If you must change the stripe size (in spite of this advice), do not make it smaller than 1/2 MB.

For effective striping, it is important to order the OSTs correctly (see Section 1.5.2, page 11).

1.4.1.1 Configuration and Performance Trade-off

For maximum aggregate performance, it is important to keep all OSTs in an LOV occupied. Consider the following circumstances when striping your file system:

- When many clients in a parallel application are each creating their own files, and where the number of clients is significantly larger than the number of OSTs, the best aggregate performance is achieved when each object is put on only a single OST.
- At the other extreme, for applications where multiple processes are all writing to one large file, it is better to stripe that single file over all of the available OSTs. Similarly, if a few processes write large files in large chunks, it is a good idea to stripe over enough OSTs to keep the OSTs busy on both the write and the read path.

1.4.1.2 Striping Example

Generally you stripe files when you create your Lustre configuration and set up your OSTs. However, you can create and stripe files at any time with the `lfs` command (see Appendix A, page 19). The following example creates the file, `npf`, with a 2 MB (`2097152` bytes) stripe that starts on OST0 (`0`) and stripes over two OSTs (`2`):

```
$ lfs setstripe npf 2097152 0 2
```

The first two megabytes, bytes `0` through `2097151`, of `npf` are placed on `OST0`, and then the third and fourth megabytes, `2097152-4194303`, are placed on `OST1`. The fifth and sixth MB are again placed on `OST0`.

1.4.2 Lustre Layout

Figure 2, page 9 shows a conceptual picture of the Lustre file system layout. It is set up during system installation. Only `-5` and `-6` nodes have PCI/x channels that allow them to be configured for Lustre. The host port adapters are each mapped to a single LUN, which is mapped to a RAID component. From a software view, each LUN is seen as a single fibre channel device and is assigned a device name such as `/dev/sda`.

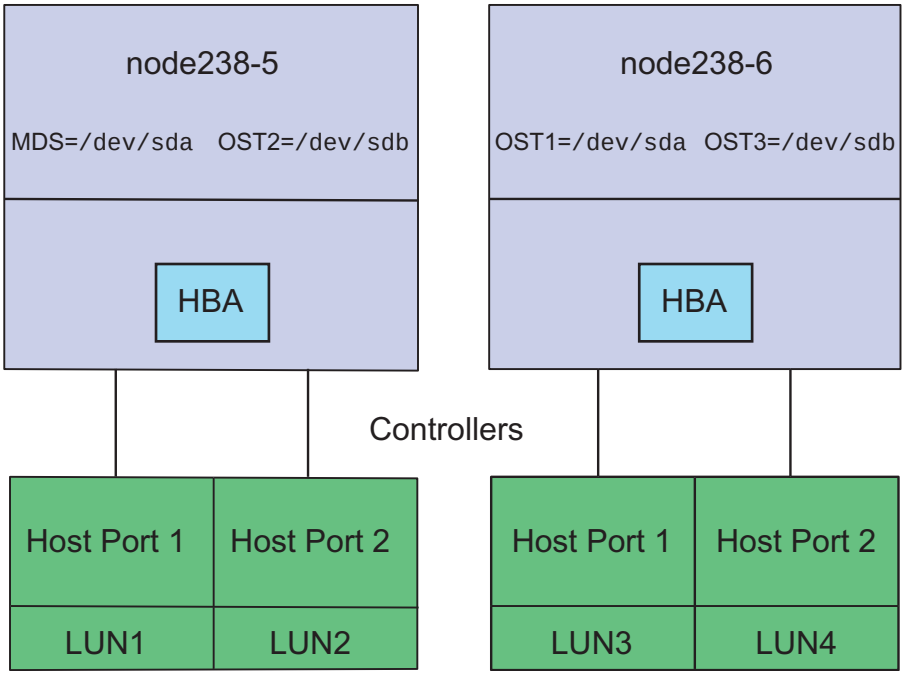

Figure 2. Lustre Node Allocation

When you construct the configuration file (see Section 1.5.2, page 11), you can set up the MDS and OSTs so that devices on the same controller are not accessed sequentially. This ensures that striping will occur across controllers. Table 2 shows an efficient pattern for entries in the `config.xml` file for a Lustre configuration in a system where you are creating and MDS and three OSTs on two nodes.

Table 2. Suggested OST Ordering

Lustre Component	Node	Device
MDS	node238-5	/dev/sda
OST1	node238-6	/dev/sda
OST2	node238-5	/dev/sdb
OST3	node238-6	/dev/sdb

See Section 1.5.2, page 11 for an example that creates one MDS and three OSTs.

1.5 Setting up Lustre

This section describes the steps you perform to create and start the Lustre file system. The overview illustrates the general procedure. Details follow.

1.5.1 Overview

To set up Lustre, perform the following activities:

1. Modify your shell script (in this example, `config.sh`) to include the `lmc` commands that create the Lustre configuration file (see Section 1.1.4.1, page 4).
   a. Remove the old configuration file, if necessary.
   b. Designate the nodes to be used by the MDS and OSTs and specify network information.
   c. Create a generic zero-configuration client.
   d. Create an MDS.
   e. Create an LOV.
   f. Create the OSTs and associate them with the LOV and the local mount point.
   g. Create the generic zero-configuration client mount point, `/mnt/lustre`.
2. Execute the shell script to create the configuration file.
3. Modify the `/etc/modprobe.conf.local` file to include Lustre-specific information.
4. Run the `lconf` command to start the OSTs and the MDS. You must start the OSTs before you start the MDS.
   a. Run the `lconf` command on each node that provides OSTs only to start it.
   b. Run the `lconf` command on the node that provides the MDS and possibly some OSTs to start the them.
   c. Mount the zero-configuration client.

1.5.2 Setup Process and Example

The example in this section modifies the shell script to create a configuration file that sets up three OSTs, an MDS, and a zero-configuration client. On a running system, you would probably configure ten or more OSTs.

The shell script is `config.sh`. The Lustre configuration file that is created when this shell script is run is named `/etc/lustre/config.xml`. It sets up the file system illustrated in Figure 2, page 9. In this configuration, the MDS is on `node238-5`, `/dev/sda`. OST1 is on `node238-6`, `/dev/sda`. OST2 is on `node238-5`, `/dev/sdb`. OST3 is on `node238-6`, `/dev/sdb`.

Note: For Active Manager to propagate the files correctly, the configuration file must be located in `/etc/lustre/config.xml` and the mount point must be `/mnt/lustre`. If your mount point is at another location, you must copy the configuration file to each node that will run MDSs or OSTs.

This example illustrates how to configure and mount a generic zero-configuration client. You could also configure and mount individual clients if you wanted.

Perform the following steps to set up the Lustre configuration as described:

1. Modify your shell script, `./config.sh`, to include the `/usr/sbin/lmc` commands that create the Lustre configuration file and define the Lustre components. Include command lines to do the following:

   a. Remove the old configuration file, if necessary. (New configuration information is appended to the old file and the complete file is accessed for Lustre operations.)

      ```
      rm -f /etc/lustre/config.xml
      ```

   b. Designate the nodes to be used by the file system and specify network information. This example configures the nodes that will be used for an MDS and three OSTs.

      In the following commands, the `--add net` option specifies a network type to the file system, the `--node` option names the host, the `--nid` option specifies the node ID, and the `--nettype` option specifies the RapidArray network.

```
lmc -m /etc/lustre/config.xml --add net --node node238-5 --nid node238-5 --nettype ra \
     --timeout 200
lmc -m /etc/lustre/config.xml --add net --node node238-6 --nid node238-6 --nettype ra \
     --timeout 200
```

c. Create a generic zero-configuration client (see Section 1.1.4.3, page 5).

```
lmc -m /etc/lustre/config.xml --add net --node client --nid '*' --nettype ra
```

d. Create an MDS called `mds1`. The `--failover` option acts as a retry flag for tasks that have timed out on the MDS. If you omit this option, the clients do not retry following a timeout and instead return I/O errors to the application.

```
lmc -m /etc/lustre/config.xml --add mds --node node238-5 --mds mds1 --fstype ldiskfs\
     --dev /dev/sda --journal_size 400 --size 500000 --failover
```

e. Create an LOV called `lov1`. Creating an LOV is a two-part process. Use the `--lov` option to associate the LOV with the MDS and then again in the commands to create the OSTs. In this example, the LOV is striped.

```
lmc -m /etc/lustre/config.xml --add lov --lov lov1 --mds mds1 --stripe_sz 1048576 \
     --stripe_cnt 0
```

f. Create the OSTs and associate them with the LOV and the local mount point. Cray recommends that you make all the OSTs the same size and that you do not mix storage types. The `--failover` option acts as a retry flag for tasks that have timed out on the OSTs . If you omit this option, the clients do not retry following a timeout and instead return I/O errors to the application.

Caution: The order of your `lmc` commands is important. Striping occurs in the order of the `lmc` declarations. For best performance, declare each OST in successive `lmc` commands to be on different nodes, then stripe on different fibre channels of the nodes.

```
lmc -m /etc/lustre/config.xml --add ost --node node238-6 --ost ost1 --lov lov1 \
     --fstype ldiskfs --dev /dev/sda --journal_size 400 --size 4000000 --failover
lmc -m /etc/lustre/config.xml --add ost --node node238-5 --ost ost2 --lov lov1 \
     --fstype ldiskfs --dev /dev/sdb --journal_size 400 --size 4000000 --failover
lmc -m /etc/lustre/config.xml --add ost --node node238-6 --ost ost3 --lov lov1 \
     --fstype ldiskfs --dev /dev/sdb --journal_size 400 --size 4000000 --failover
```

Specifying an LOV is optional. If you do not specify an LOV, the OST is associated with a default LOV.

g. Create the generic zero-configuration client mount point, `/mnt/lustre`.

```
lmc -m /etc/lustre/config.xml --add mtpt --node client --path /mnt/lustre --mds mds1 \
     --lov lov1
```

2. Execute the shell script that you have just modified.

   # **./config.sh**

   If you want to echo the activity as the configuration file is created, enter `#!/bin/bash set-x` as the first executable statement of your script. Check to see if the file is present by entering:

   # **cat ./config.xml**

3. Add the following lines to the end of the `/etc/modprobe.conf.local` file to allow the zero-configuration client mounting to work. Copy these lines to partmaster and to all nodes that are already in the partition.

```
alias lustre llite
install kptlrouter /sbin/modprobe portals; /sbin/modprobe --ignore-install kptlrouter
install ptlrpc /sbin/modprobe kranal; /sbin/modprobe --ignore-install ptlrpc
install lov /sbin/modprobe osc; /sbin/modprobe --ignore-install lov
install llite /sbin/modprobe lov; /sbin/modprobe --ignore-install llite
remove llite /sbin/modprobe -r --ignore-remove llite; /sbin/modprobe -r osc;\
      /sbin/modprobe -r kranal
```

4. Copy the configuration file to partmaster so Lustre is propagated to every OSS. The file is propagated automatically to other nodes in the system as long as it is in the `/etc/lustre` directory and is named `config.xml`.

5. Run the `lconf` command to start the OSTs and the MDS. You must start the OSTs before you start the MDS. Therefore, you run the `lconf` command on nodes that do not contain the MDS first.

   a. Run the `lconf` command on each node that provides an OST to start it.

   Warning: If this is the first time you are running the OST, you must run the `lconf` command with the `--reformat` option to reformat the device and start Lustre services on the MDS and OST nodes. Thereafter, use the `lconf --reformat` option with extreme care. Once the OST contains data, reformatting causes unrecoverable data loss, and the `lconf` command does not prompt for confirmation before reformatting.

```
# ssh node238-5
node238-5# lconf --reformat /etc/lustre/config.xml
# ssh node238-6
node238-6# lconf --reformat /etc/lustre/config.xml
```

or

```
# ssh node238-5
node238-5# lconf /etc/lustre/config.xml
# ssh node238-6
node238-6# lconf /etc/lustre/config.xml
```

Note: Be certain all OSTs, except those on a node that also contains an MDS, are started before you start the MDS.

b. Once you are certain that the OSTs are started, run the `lconf` command on the node that provides the MDS to start the MDS.

Warning: If this is the first time you are running the MDS, you must run the `lconf` command with the `--reformat` option to reformat the device and start Lustre services on the MDS and OST nodes. Thereafter, use the `lconf --reformat` option with extreme care. Once the MDS contains data, reformatting causes unrecoverable data loss, and the `lconf` command does not prompt for confirmation before reformatting.

```
# ssh node238-5
node238-5# lconf --reformat /etc/lustre/config.xml
```

or

```
# ssh node238-5
node238-5# lconf /etc/lustre/config.xml
```

c. Mount the zero-configuration client on the nodes for which you want the file system available. You do not have to specify each client node. option allows file locking through the `fcntl` system call.

```
# mount -t lustre -o nettype=ra \
 hostname:/node238-5/client /mnt/lustre
```

If you are restarting Lustre and have not modified the `lmc` commands in your script, start at step 5 because you do not need to recreate or redistribute the Lustre XML file.

Note: If the previous runtime instance of a Lustre configuration on the same nodes was not shut down cleanly, executing the `lconf` commands may initiate a recovery procedure (see Section 1.12, page 17) that may take considerable time. For more information about stopping Lustre, see Section 1.9, page 16.

1.6 Checking Lustre

To obtain information about a Lustre configuration, from the node you want to check, enter the Linux `df` command:

```
xd1-670-5$ df -t lustre
Filesystem             Type     1K-blocks    Used    Available  Use% Mounted on
xd1-670-5:/mdsa/client lustre   6332816064   2035328 6009092208   1% /mnt/lustre
```

1.7 Setting Up User Home Directories

Lustre is optimized for large file I/O. If you want to create home directories on a Lustre file system, Cray recommends that you do not put them in the same Lustre file system as the one intended for large I/O.

To configure home directories to work over Lustre, first configure the zero-configuration client to mount on `/mnt/lustre` (see the example in Section 1.5.2, page 11). Next configure the home directories to link to Lustre by performing the following steps:

1. Be sure that no users are logged in. Close all partitions and wait for jobs to complete and users to log out.
2. Go to the `/var/opt/pce` directory and copy the contents of the existing home directories to the Lustre file system:

   ```
   # cd /var/opt/pce
   # cp -ipr home /mnt/lustre
   ```

3. Move or remove the existing directories so they do not interfere with the link, then link to the Lustre home directory:

   ```
   # mv -i home home.bak
   ```

   or

   ```
   # rm -rf home
   # ln -s /mnt/lustre/home /home
   ```

4. Create an entry in the `/etc/fstab` file to mount the home directory.

1.8 Increasing Data Storage

You can increase data storage in your Lustre file system by saving your data, recreating your configuration with more OSTs, and copying your data back.

The maximum size of a Lustre volume is two terabytes times the number of OSTs.

1.9 Stopping Lustre

To stop Lustre, reverse the steps you performed to start it. The following example stops the file system where node `node238-5` (`/dev/sda`) is the MDS and nodes `node238-5` (`/dev/sdb`) and `node238-6` (`/dev/sda` and `/dev/sdb`) are the OSTs.

1. Unmount Linux Lustre clients.

```
# ssh node285
node285# umount /mnt/lustre
node285# ssh node291
node291# umount /mnt/lustre
node291# ssh node274
node274# umount /mnt/lustre
```

2. Run the `lconf -d` command to shut down the MDS.

```
# ssh node238-5
node238-5# lconf -d /etc/lustre/config.xml
```

3. Run the `lconf -d` command to shut down the OSTs by node.

```
# ssh node238-6
node238-6# lconf -d /etc/lustre/config.xml
node238-6# ssh node238-5
node238-5# lconf -d /etc/lustre/config.xml
```

You do not need to remove the script and Lustre configuration file.

You can also run the `lconf -f` command to force shut down. However, if you do this, your restart time may be long.

Caution: Shutting down the MDS before the OSTs minimizes the chances of needing Lustre recovery. If not all clients have terminated, there may be I/O in the buffers. If you cannot shut down nodes gracefully, you may lose data.

1.10 Debugging the File System

To facilitate debugging the Lustre file system and help Cray service personnel, capture Linux `dmesg` or `syslog` information as soon as possible after failure. Contact your Cray service representative.

1.11 Collecting Lustre Log Files

When Lustre encounters a problem, dump files are generated on the MDS and OSS nodes in log files in the `/tmp` directory. Log files are named by a timestamp and pid, for example:

```
/tmp/lustre-log-node230-5.1122323203.645
```

There may be many files in each `/tmp` directory or there may be none. Because the files reside in the `/tmp` directory, they will disappear on reboot.

Create a script to retrieve the dump files and store them in the location where other system logs are kept. You can `tar` these dump files.

1.12 Recovery

If you do not cleanly shut down the Lustre servers before you restart, Lustre may automatically enter a recovery process to restart. There is no warning and recovery may take a long time.

Note: You must let the recovery process continue; it will restart Lustre.

Other failures that can trigger recovery include:

- Client node failure
- MDS failure
- OST failure
- Transient network failure

After a crash a client state can persist if the client exists and is operable when the server is brought up. Upon recovery, the MDS allows a default time period for clients to contact it to replay incomplete transactions. When the transactions are completed, these clients are reauthorized as current valid clients. When the default time period expires, the MDS accepts connections and requests from new clients as well as new instances of old clients.

1.13 Troubleshooting

The following sections help you troubleshoot some of the problems affecting your file system. Because typographic errors in your configuration script or your shell script can cause many kinds of errors, check these files first when something goes wrong.

1.13.1 identifying MDSs and OSTs

Run the `lfs check servers` command to identify the OSTs and MDS for the file system. You must be root user to run this command.

```
# lfs check servers
```

If there is more than one Lustre file system, the `lfs check servers` command does not sort the OSTs and MDSs. You must do this yourself.

You can then check the status of individual nodes with the `lfs find command:`

```
# lfs find /lus/node207-5 | grep ACTIVE
```

You can write a script to recursively look at the nodes.

1.13.2 Hung Nodes

There is no way to clear a hung node except by rebooting. Unmount the clients, shut down the MDS and OSTs, and shut down the system. For more information, see Section 1.9.

1.13.3 Correcting Unbalanced Server Activity

If you notice that all activity is taking place on one server before moving to another, check the OSTs and striping pattern defined in the Lustre configuration file to be sure that your stripe pattern takes the location of the OSTs into account.

Run the Linux `lsscsi` command to determine the LUNs that each OST accesses. For example, if OST1 and OST2 are on the same server, these two stripe to the same device.

For example, looking at `node238-5`:

```
node238-5:~ # lsscsi
[0:0:0:0]    disk    DDN      S2A 8500         5.12  /dev/sda
[0:0:0:1]    disk    DDN      S2A 8500         5.12  /dev/sdb
```

Lustre Configuration Management Tools [A]

The Lustre make configuration utility, `lmc`, generates an XML configuration file given data describing the system. You do not generally run the `lmc` command directly.

The Lustre configuration utility, `lconf`, provides low-level configuration based on the XML configuration file generated by the `lmc` command.

The stripe utility, `lfs`, allows you to stripe files once Lustre has been configured. Most often, you stripe files by supplying options to the `lmc` command. For more information about setting up Lustre, see Section 1.5, page 10 .

A.1 `lmc`

The `lmc` utility provides commands to generate a Lustre configuration file in XML format. The utility has the form:

```
lmc [options]
```

The options of `lmc` that you are most likely to use include:

`-o` *filename*
`--output` *filename*

Writes XML configuration into given output file; overwrites existing content.

`-m` *filename*
`--merge` *filename*

Appends to the specified configuration file.

`--stripe_sz` *arg*

Specifies the stripe size in bytes.

`--stripe_cnt` *arg*

Specifies the number of OSTs each file should be striped on. (default = `0` means stripe across one OST)

`--stripe_pattern` *arg*

Specifies the stripe pattern. RAID 0 is the only one currently supported. (default = `0`)

`--add node` *arg*

Adds a node with the specified characteristics to the cluster.

`--add net` *arg*

Adds a network device with the specified characteristics to the cluster.

`--add mds` *arg*

Adds the MDS with the specified characteristics to the cluster.

`--add lov` *arg*

Adds an LOV with the specified characteristics to the cluster.

`--add ost` *arg*

Adds an OST with the specified characteristics to the cluster.

`--add mtpt` *arg*

Creates a mount point on the specified node.

`--nid` *arg* Specifies the node ID number.

`--node` *arg* Adds a host name to the cluster configuration.

`--nettype` *arg*

Specifies the network type.

`--mds` *arg* Specifies the MDS name.

`--lov` *arg* Specifies the LOV name.

`--ost` *arg* Specifies the OST name.

`--add` *arg* Adds the component to the configuration.

`--fstype` *arg*

Specifies the file system type. For Lustre 2.6 kernels, the `ldiskfs` filesystem must be used.

`--journal_size` *arg*

Specifies the new journal size for `ext3` file system.

`--dev` *arg* Specifies the path of the device on local system.

`--size` *arg* Specifies the size of the device in KB.

`--path` *arg* Specifies the mount point for Lustre.

`--clientoptions` *arg*

Specifies the options for Lustre, such as async.

`--timeout` *arg*

Sets the timeout to initiate recovery.

To identify other `lmc` options, type:

```
$ lmc -h
```

at the command line prompt.

A.2 lconf

The `lconf` utility configures a Lustre node using the XML configuration file created by the `lmc` command. It performs all the steps to configure the node, including loading the kernel modules and preparing block devices, if necessary. The utility requires the name of the XML configuration file and has the form:

`lconf [options]` *config.xml*

The options of `lconf` that you are most likely to use include:

config.xml Specifies the Lustre configuration file to be used by `lconf`.

`--node` *nodename*

Loads the configuration for *nodename*.

`-d`
`--cleanup`

Stops the configuration on node on which command is run.

`-f`
`--force` Forces the Lustre file system to unmount.

`-n`
`--noexec` Prints the steps that the `lconf` will perform without executing them.

`--inactive` Ignores the named OST during the Lustre mount.

`--reformat` Reformats all devices. This is essential the first time the system is brought up.

To identify other `lconf` options, type:

```
$ lconf -h
```

at the command line prompt.

A.3 `lfs`

The `lfs` utility lets you create a new file and define its striping pattern, determine the default striping pattern of a file, or gather the extended attributes (object numbers and location) of a file. The command can be run with arguments or interactively if you do not specify them. Type `exit` or `quit` to end an interactive session.

To create a new file with a specific striping pattern, to set the default striping pattern on an existing directory, or to delete the default striping pattern (`-d` option) from an existing directory, enter the following arguments:

`lfs setstripe` *`filename|dirname stripe_size stripe_start stripe_count`*

or

`lfs setstripe -d` *`dirname`*

where

filename or *dirname*

Names the file or directory on which to operate.

stripe_size

Specifies the number of bytes in each stripe. (default = 0)

stripe_start

Specifies the OST index of the first stripe. (default = -1)

stripe_count

Specifies the number of OSTs to stripe over. (default = 0; all = -1)

-d *dirname* Deletes striping in the named directory.

See Section 1.4.1.2, page 8 for an example of setting a stripe.

To list the extended attributes for a given file, for files in a directory, or for all files in a directory tree, enter the following arguments:

```
lfs find [--obd uuid] [--quiet | --verbose] [--recursive] filename|dirname
```

where

filename or *dirname*

Names the file or directory on which to operate

uuid Specifies the universal unique identifier.

`--quiet` Specifies minimal output.

`--verbose` Displays output operations as they progress.

`--recursive`

Proceeds through directory tree

To list the extended attributes of file `skf`, enter:

```
$ lfs find /mnt/lustre/skf
OBDS
   0:OST_localhost_UUID
/mnt/lustre/skf
obdidx    objid   objid   group
0         1       0x1     0
```

To recursively list the extended attributes of all files in the `/mnt/lustre` directory, enter:

```
$ lfs find -r /mnt/lustre
```

To list all files that have objects on host `OST2_UUID`, enter:

```
$ lfs find -r --obd OST2_UUID /mnt/lustre
```

To list the striping pattern for filename `pkg`, enter the following argument:

```
lfs getstripe pkg
```

where

filename Names the file on which to operate.

To display the status of specified MDS or OSTs or of all the servers (MDS and OSTs), enter the following argument:

`lfs check` *`osts`*|*`mds`*|*`servers`*

where

osts Displays the status of all OSTs.

mds Displays the status of the MDS.

servers Displays the status of all MDS and OSTs.

For example to check the status of all servers, that is the MDS and OSTs, enter:

```
$ lfs check servers
OSC_localhost.localdomain_OST_localhost_mds1 active.
OSC_localhost.localdomain_OST_localhost_MNT_localhost active.
MDC_localhost.localdomain_mds1_MNT_localhost active.
$
```

To identify other `lfs` options, type:

```
$ lfs -h
```

at the command line prompt.

Sample Configuration File [B]

The configuration file shown below is provided as a sample. Modify the text to match your system layout. This example configures an MDS and three OSTs.

For more information about configuration files, see Section 1.1.4.1, page 4.

```
#!/bin/bash

config=${1:-xd1-lustre.xml}

SERVERS="node238-5 node238-6"
JSIZ='--journal_size 400'   # suggested ext3 journal size
LMC=${LMC:-/usr/sbin/lmc}
STRIPE_BYTES=1048576

if [ -f $config ]
then
  rm -f $config
fi

# create network node for each server

for s in $SERVERS
do
 ${LMC} -m $config --add net --node $s --nid $s --nettype ra || exit 1
done

#add network node for a generic client

${LMC} -m $config --add net --node client --nid '*' --nettype ra || exit 1

# configure mds server
${LMC} -m $config --add mds --node node238-5 --mds mds1 --group mds --failover  \
     --fstype ldiskfs --dev /dev/sda $JSIZ --timeout 300 || exit 10

# configure lov
#
${LMC} -m $config --add lov --lov lov1 --mds mds1 --stripe_sz $STRIPE_BYTES \
     --stripe_cnt 0 || exit 20

# configure osts
```

```
#
${LMC} -m $config --add ost --node node238-6 --ost ost1 --group ost1 --failover \
     --lov lov1 --fstype ldiskfs --mountfsoptions extents,mballoc --dev /dev/sda $JSIZ \
     --timeout 300 || exit 21

${LMC} -m $config --add ost --node node238-5 --ost ost2 --group ost2 --failover \
     --lov lov1 --fstype ldiskfs --mountfsoptions extents,mballoc --dev /dev/sdb $JSIZ \
     --timeout 300 || exit 21

${LMC} -m $config --add ost --node node238-6 --ost ost3 --group ost3 --failover \
     --lov lov1 --fstype ldiskfs --mountfsoptions extents,mballoc --dev /dev/sdb $JSIZ \
     --timeout 300 || exit 21

# create generic zero-configuration mount point
#
${LMC} -m $config  --add mtpt --node client --path /mnt/lustre --mds mds1 \
     --lov lov1 --clientoptions async || exit 30
```

Glossary

Active Manager

The software that monitors and manages all aspects of the Cray XD1 system. Its user interfaces provide administrators and end users with a single point of control for the system.

administrator

A user of the Cray XD1 system with unlimited access privileges, including permission to issue all Active Manager commands. The administrator is responsible for monitoring and managing the system.

Cray XD1 system

A stand-alone Cray XD1 chassis or multiple chassis that communicate over both the supervisory network and the RapidArray interconnect.

force

An option on some Active Manager commands; specifies that the command must execute even though it may have disruptive results.

link

See *RapidArray link*.

logical object volume (LOV)

A grouping of (Lustre file system) OSTs that are accessed as a unit.

Lustre configuration file

An XML file that specifies how the Lustre file system is structured. It is built by modifying the shell script to include lmc (Lustre make configuration) commands that define the file system components.

metadata server (MDS)

The component of the Lustre file system that stores file metadata.

node

An instance of the Linux operating system and the hardware components that it controls. The hardware components in a Cray XD1 node include an SMP and its associated memory, one or two RapidArray processors (depending on configuration) and, optionally, an FPGA application acceleration processor.

node ID

The hardware-based identifier of a node, which has the following form: "chassis-id.node-ordinal" where "chassis-id" is the chassis ID and "node-ordinal" is the number of the node within the specified chassis (nodes are numbered from 1 to 6, left to right as viewed from the front of the chassis).

object storage server (OSS)

A node that hosts the OST(s) of the Lustre file system.

object storage target (OST)

The component of the Lustre file system that handles file activities.

partition

A logical group of nodes with the same operating system version and configuration; may reside in more than one Cray XD1 chassis. Partitions enable an organization to dedicate a set of nodes to perform a particular function (run a type of job, host a system-wide service, or serve a particular user group). Users treat the set of nodes in a partition as a single, homogeneous computing resource. Administrators specify the attributes of a partition. See also *partition master software image*.

partition master software image

The software image associated with a partition; used to generate the working software images of nodes that are allocated to the partition. The partition master software image is created from a combination of an application release master, a configuration determined by the partition's attributes, any other partition-wide configuration (such as services), and any installed local or third-party software.

RapidArray link

The physical communication path between two RapidArray ports. Each link can carry two gigabytes per second.

zero-configuration client

A generic client that allows you to mount the Lustre file system on any Linux node.

Index